DSC 461 Project - Job Application Database

Milestone 1

Team: Fastest HK Journalist (# 11)

Members: Zhou Xu (zxu17), Kefu Zhu (kzhu6), Hao Jiang (hjiang23)

Problem Statement

With the rapid development of internet, web-based job applications become ubiquitous in the United States. However, as the job application data increases in size, many companies encounter problems due to lack of efficient database systems. For example, the application status cannot be updated in a timely manner or contradictory entries were input from multiple ends. These problems confuse users and always lead to great loss in forms of time and money, which lowers the efficiency of company operations.

The relational database system that we are developing will bring a solution to manage complex data of job applications for a company and provides an user-friendly platform for HR, employees and applicants to insert, retrieve and modify their portion of data with permission. In addition, the database will hold information for advertisement platforms that each job is posted. It will help collect and structure data so that further analysis and decisions could be made based on the frequency each platform is used.

The database system is better than traditional spreadsheet tools such as excel from the following perspectives:

- **Date integrity**: Domain constraints can be added to restrict the type of data under each attribute for each entry. This will prevent unintentional incorrect data input on some level.
- **Consistency**: Only one version of data copy can exist at any given moment, which eliminates the ambiguity of information.
- **Version control**: Versions of a database are better managed and the data in a database can be easily recovered.
- **Readability**: Users can define their favorable ways of retrieving the data using queries.
- **Visibility**: A database can have data visible to all users with relevant permissions.
- **Scalability**: A database can work with a greater amount of data without worrying about the limits.
- **Productivity**: End users can rely on the embedded algorithms, which increases the efficiency of workflows.

- **Extensibility**: A database can be integrated with UI and different tools, which better prepares the structure of data for further analysis.
- **Collaborativity**: Users can work from multiple ends concurrently.
- **Security**: Users are assigned with different level of permission, which improves the security.
- **Redundancy**: A database stores data in one place to minimize the redundancy and it only takes up the space it needs.

Target users

Users of this database include employees (especially HR) of the company as well as job applicants and headhunters who are looking for jobs. The administrator of the database will be the data engineer in the IT department of the company.

List of Relations

1. **Office_Location**
    a. OfficeID: The identification number for each office
    b. StreetAddress: The street address of the office
    c. City: the city name where the office is located
    d. State: the state name where the office is located
    e. ZipCode: the zip code of the office

    Note:

    1. The primary key of this relation is OfficeID
    2. The StreetAddress has UNIQUE constraint

```
+---------------+-------------+------+-----+---------+-------+
| Field         | Type        | Null | Key | Default | Extra |
+---------------+-------------+------+-----+---------+-------+
| OfficeID      | int(5)      | NO   | PRI | NULL    |       |
| StreetAddress | varchar(50) | YES  | UNI | NULL    |       |
| City          | varchar(25) | YES  |     | NULL    |       |
| State         | varchar(25) | YES  |     | NULL    |       |
| ZipCode       | int(5)      | YES  |     | NULL    |       |
+---------------+-------------+------+-----+---------+-------+
```

2. **Department**
    a. Name: The name of department

b. Dnumber: The identification number of department

c. OfficeID: The identification number of office

d. ManagerID: The employee ID of the manager of department

Note:

1. The primary key of this relation is (Name, Dnumber)
2. OfficeID is a foreign key that references to the OfficeID in Office_Location relation
3. ManageID is a foreign key that references to the EmployeeID in Employee relation

```
+-----------+-------------+------+-----+---------+-------+
| Field     | Type        | Null | Key | Default | Extra |
+-----------+-------------+------+-----+---------+-------+
| Name      | varchar(50) | NO   | PRI | NULL    |       |
| Dnumber   | int(5)      | NO   | PRI | NULL    |       |
| OfficeID  | int(5)      | YES  | MUL | NULL    |       |
| ManagerID | int(5)      | YES  | MUL | NULL    |       |
+-----------+-------------+------+-----+---------+-------+
```

3. Employee

a. EmployeeID: The identification number of employee

b. FirstName: The first name of employee

c. LastName: The last name of employee

d. Position: The position of employee

e. Department: The department where employee belongs to

Note:

1. The primary key of this relation is EmployeeID
2. Department is a foreign key that references to the Name in Department relation

```
+------------+-------------+------+-----+---------+-------+
| Field      | Type        | Null | Key | Default | Extra |
+------------+-------------+------+-----+---------+-------+
| EmployeeID | int(5)      | NO   | PRI | NULL    |       |
| FirstName  | varchar(20) | YES  |     | NULL    |       |
| LastName   | varchar(20) | YES  |     | NULL    |       |
| Position   | varchar(50) | YES  |     | NULL    |       |
| Department | varchar(50) | YES  | MUL | NULL    |       |
+------------+-------------+------+-----+---------+-------+
```

4. Open_Position

a. JobID: The identification number of open position

b. OfficeID: The identification number of office

c. Type: The type of open position (FullTime/PartTime)

d. PostDate: The date when position opens to application

e. Position: The name of open position

f. Department: The name of department which hires people for the open position

g. Salary: The annual salary for open position

h. Duration: The number of required work years for open position

i. VISASponsor: Whether the open position sponsor work visa (TRUE/FALSE)

j. AdsPlatform1: The name of advertisement platform # 1

k. AdsPlatform2: The name of advertisement platform # 2

l. AdsPlatform3: The name of advertisement platform # 3

Note:

1. The primary key of this relation is JobID
2. OfficeID is a foreign key that references to the OfficeID in Office_Location relation
3. Department is a foreign key that references to the Name in Department relation

Field	Type	Null	Key	Default	Extra
JobID	int(5)	NO	PRI	NULL	
OfficeID	int(5)	YES	MUL	NULL	
Type	char(8)	YES		NULL	
PostDate	date	YES		NULL	
Position	varchar(50)	YES		NULL	
Department	varchar(50)	YES	MUL	NULL	
Salary	int(11)	YES		NULL	
Duration	float	YES		NULL	
VISASponsor	tinyint(1)	YES		NULL	
AdsPlatform1	varchar(50)	YES		NULL	
AdsPlatform2	varchar(50)	YES		NULL	
AdsPlatform3	varchar(50)	YES		NULL	

5. Application_Received

a. ApplicationID: The identification number of submitted application

b. FirstName: The first name of the applicant

c. LastName: The last name of the applicant

d. Email: The contact email address of the applicant

e. School: The name of school where the applicant graduated

f. Degree: The highest degree obtained by the applicant

g. JobID: The identification number of the job that the applicant applied for

h. Platform: The name of platform where the applicant applied for this open position

i. RecruiterID: The employee ID of the recruiter for this applicant

Note:

1. The primary key of this relation is ApplicationID
2. JobID is a foreign key that references to the JobID in Open_Positions relation
3. RcruiterID is a foreign key that references to the EmployeeID in Employee relation

Field	Type	Null	Key	Default	Extra
ApplicationID	char(5)	NO	PRI	NULL	
FirstName	varchar(30)	YES		NULL	
LastName	varchar(30)	YES		NULL	
Email	varchar(50)	YES		NULL	
School	varchar(80)	YES		NULL	
Degree	varchar(50)	YES		NULL	
JobID	int(5)	YES	MUL	NULL	
Platform	varchar(50)	YES		NULL	
RecruiterID	int(5)	YES	MUL	NULL	

Web-interface

1. **Applicant view** (See Fig.1). This is a public job posting page that lists the current open positions of the company. Users can search using JobID or position keywords and also applying filters on the locations and departments. When clicking on a single position, it will redirect to a new page with a online application form to fill out, which functions as an insert method.
2. **HR view** (See Fig.2). This is a page that manages the job posting from the HR side inside the company. The UI structure is mostly the same, but contains more information that applicants cannot see, such as post date, salary and visa sponsorship. When clicking on a single position, it will direct to a new page showing all current applicants of this position. For both Open_Positions and Applicatoin_Received relations, tuples can be inserted, deleted and modified by clicking the icons.

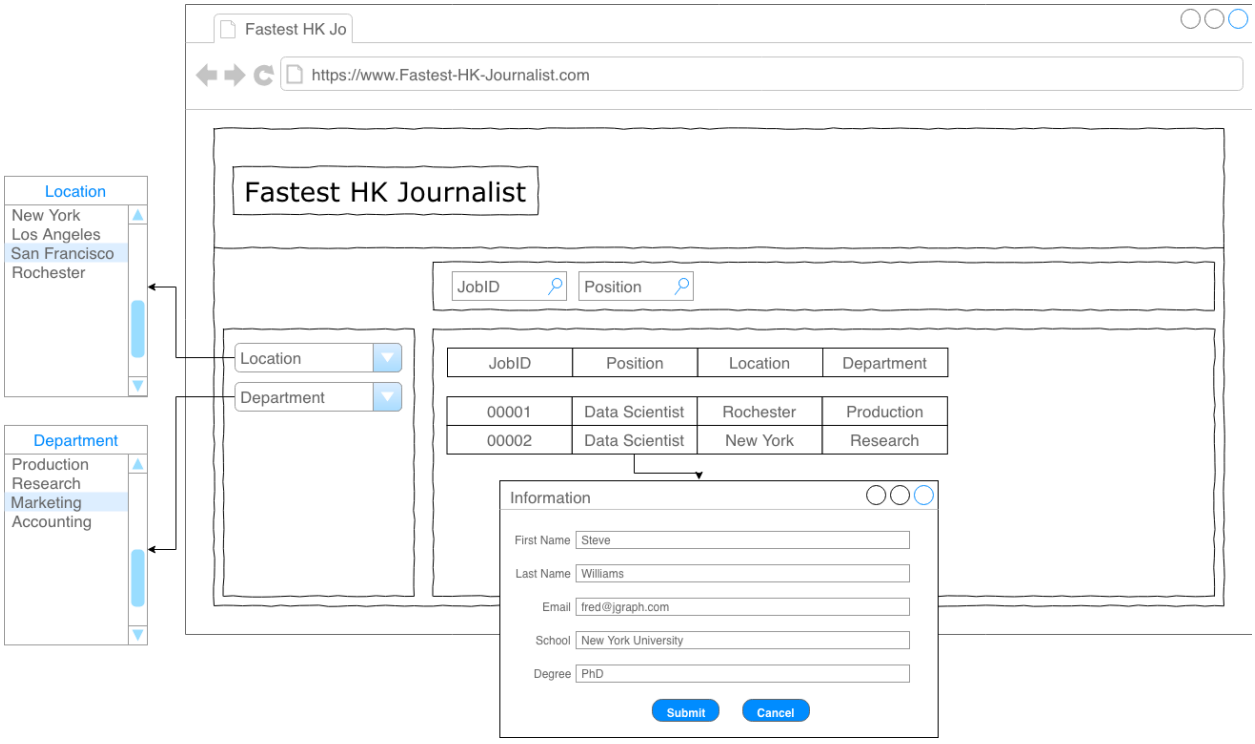

Figure 1. User Interface for Job Applicant

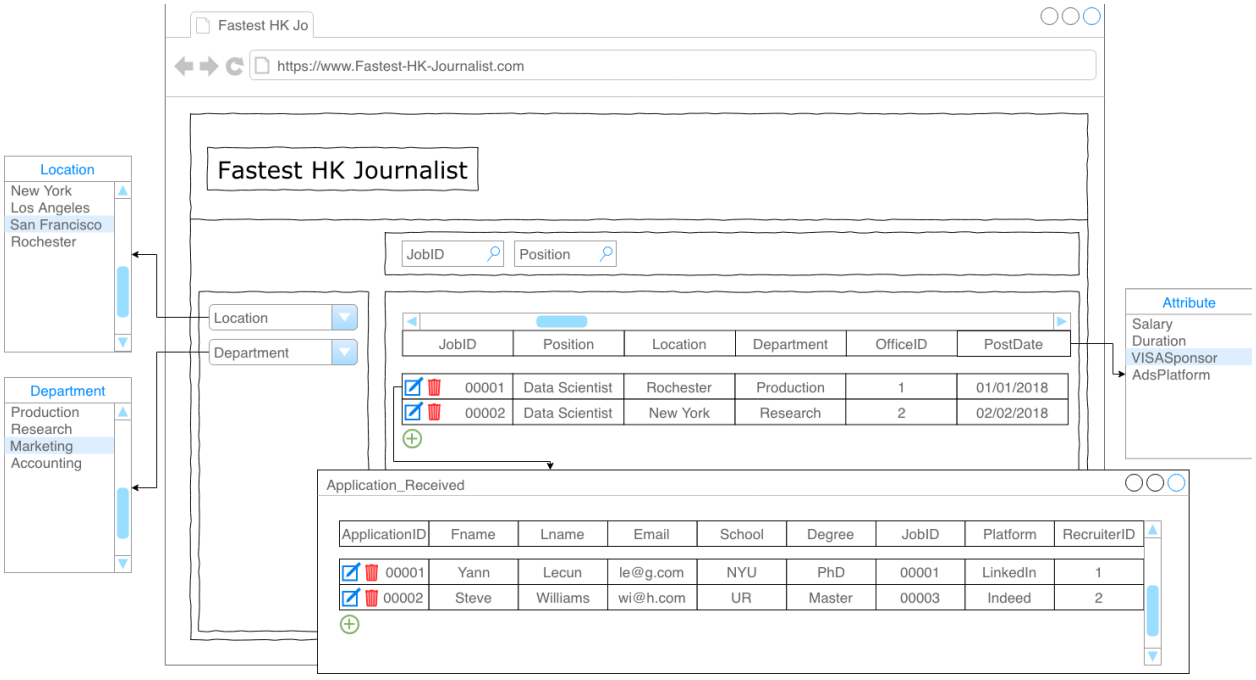

Figure 2. User Interface for Human Resource

Data

The relations will be populated using data created by ourselves while referencing to real-world data to some degree.

Future work

1. Add web interface to insert, delete and modify the Office_Location, Department and Employee relations.
2. Add user-defined domain type for **Open_Positions(Type)** and **Application_Received(Email)**
    - Alternative solution: Create a trigger for any insert or update on these two attributes
3. Modify the default value for ON UPDATE and ON DELETE for all foreign keys.